SN Computer Science (2021) 2:279
https://doi.org/10.1007/s42979-021-00658-w

ORIGINAL RESEARCH

Anomaly Detection in COVID-19 Time-Series Data

Hajar Homayouni[1] · Indrakshi Ray[1] · Sudipto Ghosh[1] · Shlok Gondalia[1] · Michael G. Kahn[2]

Received: 9 January 2021 / Accepted: 26 April 2021 / Published online: 19 May 2021

Abstract

Anomaly detection and explanation in big volumes of real-world medical data, such as those pertaining to COVID-19, pose some challenges. First, we are dealing with time-series data. Typical time-series data describe behavior of a single object over time. In medical data, we are dealing with time-series data belonging to multiple entities. Thus, there may be multiple subsets of records such that records in each subset, which belong to a single entity are temporally dependent, but the records in different subsets are unrelated. Moreover, the records in a subset contain different types of attributes, some of which must be grouped in a particular manner to make the analysis meaningful. Anomaly detection techniques need to be customized for time-series data belonging to multiple entities. Second, anomaly detection techniques fail to explain the cause of outliers to the experts. This is critical for new diseases and pandemics where current knowledge is insufficient. We propose to address these issues by extending our existing work called IDEAL, which is an LSTM-autoencoder based approach for data quality testing of sequential records, and provides explanations of constraint violations in a manner that is understandable to end-users. The extension (1) uses a novel two-level reshaping technique that splits COVID-19 data sets into multiple temporally-dependent subsequences and (2) adds a data visualization plot to further explain the anomalies and evaluate the level of abnormality of subsequences detected by IDEAL. We performed two systematic evaluation studies for our anomalous subsequence detection. One study uses aggregate data, including the number of cases, deaths, recovered, and percentage of hospitalization rate, collected from a COVID tracking project, New York Times, and Johns Hopkins for the same time period. The other study uses COVID-19 patient medical records obtained from Anschutz Medical Center health data warehouse. The results are promising and indicate that our techniques can be used to detect anomalies in large volumes of real-world unlabeled data whose accuracy or validity is unknown.

Keywords Anomaly detection · Data quality tests · COVID-19 data · Explainability · LSTM-autoencoder · Time series

Introduction

Large amounts of data records are being collected from various sources over time to analyze the immediate and long-term impacts of COVID-19 on human health. Examples include the analysis of the impacts [1], diagnosis [2], treatments [3], and pre-symptom detection [4] of COVID-19 based on the available data collected from radiography, chest CT, chest X-ray, wearable devices, and COVID-19 tracking reports. Such data records are assumed to be accurate. However, the records may get corrupted in the non-trivial data collection and transformation processes. Anomalous data may lead to incorrect inferences and research findings. Thus, it is critical to automatically find inaccurate or anomalous data before doing any analysis and explain how the data is anomalous to the healthcare professionals.

This article is part of the topical collection “Artificial Intelligence for HealthCare” guest edited by Lydia Bouzar-Benlabiod, Stuart H. Rubin and Edwige Pissaloux.

✉ Sudipto Ghosh
ghosh@colostate.edu

Hajar Homayouni
hajar.homayouni@colostate.edu

Indrakshi Ray
iray@colostate.edu

Shlok Gondalia
shlok@rams.colostate.edu

Michael G. Kahn
michael.kahn@cuanschutz.edu

[1] Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA

[2] Anschutz Medical Campus, University of Colorado Denver, Aurora, CO 80045, USA

Existing anomaly detection techniques for COVID-19 data (some based on machine learning) focus only on outbreak detection [5–8] in the COVID-19 tracking cases across the world. Machine learning approaches [9–11] have also been used for data quality assurance in other domains. Many of these techniques use supervised machine learning, which assumes the existence of labeled training data which in real-world is unavailable for new forms of outbreaks such as COVID-19. Moreover, outlier detection using machine learning fails to explain how records are anomalous to the domain experts. Finally, most works [12–17] identifying anomalous records in a data set cannot be used on time-series data as anomalies may span multiple attributes and records in a sequence [18]. We aim to eliminate the above shortcomings by detecting anomalies in COVID-19 time-series data without having access to labeled data and explaining the anomalies to domain experts in a comprehensible manner.

This approach, called IDEAL, builds upon our previous work on data quality assessment approach [19] that uses an LSTM-autoencoder [10] network to find anomalies in unsupervised data. Anomalies are data records or subsequences of data records whose behaviors (i.e., attribute values or change in the values over time) are significantly different from the majority of records and subsequences in a time-series data set [5]. IDEAL automatically (1) discovers different types of constraints from the sequence data, (2) marks subsequences and records that violate the constraints as suspicious, and (3) explains the violations. IDEAL automatically generates three types of visualizations to explain the anomalies. The plot showing the suspiciousness score per attribute indicates which attributes make the subsequence anomalous. The second visualization uses decision trees to illustrate the violated constraints. The third plot compares a suspicious subsequence detected by IDEAL with normal subsequences belonging to the same data set. The approach incorporates feedback from domain experts to improve the accuracy of constraint discovery and anomaly detection. We proposed an autocorrelation-based reshaping technique that automatically adjusts the LSTM-autoencoder input window size based on how far the records are related to their past values. We evaluated the effectiveness of IDEAL using data sets from Yahoo servers [20], NASA Shuttle [21], and Colorado State University Energy Institute [22]. We demonstrated that IDEAL could detect previously known and injected anomalies in these data sets.

The above mentioned work needs to be extended for COVID-19 time-series data. The Yahoo servers [20] and NASA Shuttle [21] data sets that we previously used contain time-series data associated with a single entity; COVID-19 time-series data belongs to multiple entities (e.g., cases and deaths for states and counties, or lab test type and test name for different patients. Thus, in COVID-19 data there are multiple subsets of data each of which belongs to a single entity. Records in each subset are temporally dependent but they are unrelated to records in other subsets. Moreover, in each subset there are multiple grouping attributes (e.g., test type and test name) which requires the data to be pre-processed to make the results correct. We extend IDEAL using a two-level reshaping approach to transform data into a shape that is suitable for analysis. This approach removes the restriction that all data records in a sequence data set must be temporally dependent and are describing behaviors of the same object over time. Instead, IDEAL supports data sets in which a subset of records are temporally related to each other but are unrelated to the records from other subsets in the same data set. For example, a health data store may contain medical records of multiple patients over time. The records of each patient are temporally dependent but independent from those of other patients.

One naïve solution may be to directly split the data based on grouping attribute(s) and generate multiple temporally-dependent subsequences. However, such a solution does not preserve associations among grouping attribute values. Consequently, IDEAL uses a pivoting-based approach to split data into multiple independent subsequences in a manner that preserves the associations among grouping attribute values.

In addition, IDEAL also offers an explanation of the anomalous behavior. It provides a data visualization plot that explains the level of abnormality of anomalous subsequences detected by the approach. This plot visualizes the data over time to help a domain expert understand the difference between the attribute values of a suspicious subsequence with those of other subsequences in the data set. Such a plot draws attention to the anomalous subsequences; this is especially useful for large volumes of data where there is a lack of domain knowledge.

We conduct two types of studies to evaluate the anomaly detection effectiveness of IDEAL in the absence of domain knowledge. The first study validates the level of abnormality of an anomalous subsequence generated from a data source that is detected by IDEAL by comparing it with subsequences generated from other data sources that are reporting the same information. The data sources we use to conduct this study are COVID-19 tracking data collected from Johns Hopkins [23], New York Times [24], and COVID Tracking project [25] repositories. We demonstrate that IDEAL can detect anomalous subsequences which are indeed outliers when compared with other data sets reporting the same information.

The second study validates the level of abnormality of the suspicious subsequences by comparing the suspicious subsequences detected by IDEAL against other subsequences generated from the same data set. Here the data sets correspond to a homogeneous population, i.e., a phenotype of people with the same values for personal (e.g., gender and age) and medical (e.g., diagnosis category, disease type, and medication type) attributes. We use COVID-19 medical data collected from the health data warehouse in Anschutz medical campus [26]. We demonstrate that IDEAL can detect abnormal subsequences from the data sets under test.

The contributions of this work are as follows:

- We propose a two-level reshaping technique to prepare data for training the LSTM-autoencoder model. Thus, the preprocessing step that we develop allows IDEAL to be used on different types of time-series data, possibly grouped by different attributes, and belonging to multiple entities.
- We propose a data visualization plot to explain the level of abnormality of the subsequences detected by IDEAL. This helps the domain experts quickly identify the anomalous portions of data in large data sets.
- We propose systematic validation techniques based on a comparison between suspicious and other subsequences to demonstrate the anomaly detection effectiveness of IDEAL. Such a method is useful when there is a lack of labeled data or where there is insufficient domain knowledge.

The value of this work lies in automating the process of detecting and explaining potential anomalies that allow clinicians who have domain knowledge but lack data science skills to evaluate the effect of the level of abnormality and the seriousness of an anomaly on the clinical research question they are seeking to answer from the COVID-19 data. Due to the large number of investigators who intend to use the COVID-19 data, the use of the approach could potentially benefit a wide range of clinical investigators. This work can also be used for other domains that are analyzing large volumes of unlabeled time-series data that belong to multiple entities.

The rest of the paper is organized as follows. “Related work” describes the related work. “Approach” provides an overview of IDEAL. “Extension to data preparation” describes how we handle time-series data belonging to multiple independent objects. “Extension to anomaly interpretation” discusses anomaly interpretation in depth. “Evaluation” presents the evaluation of our approach. “Conclusions” concludes the paper and outlines directions for future work. Appendix A explains the architecture of the LSTM-autoencoder network.

Related Work

The existing anomaly detection techniques in COVID-19 data focus only on outbreak detection [5–8] in the COVID-19 tracking cases across the world. Karadayi et al. [5] used a hybrid autoencoder network composed of a 3D convolutional neural network (CNN) and an autocorrelation based network for outbreak detection from spatio-temporal COVID-19 data provided by the Italian Department of Civil Protection. Jombart et al. [6] used linear regression, generalised linear models (GLMs), and Bayesian regression to detect sudden changes in potential COVID-19 cases in England. However, there has been no focus on quality assurance of COVID-19 data used for various analysis.

Machine learning-based techniques used for outlier detection in non-sequence data, such as support vector machine (SVM) [12], local outlier factor (LOF) [13], isolation forest (IF) [14], and elliptic envelope (EE) [15] have been used to detect anomalous records from time series data [27]. Such approaches do not consider temporal dependencies between data records and can only detect trivial out-of-range outliers.

Techniques that detect anomalous records from time-series data can be categorized as decomposition and modeling techniques. Decomposition techniques, suitable only for univariate time series, break a time series into level, trend, seasonality, and noise components and monitor the noise components to capture the anomalous records [28, 29]. Modeling techniques represent a time series as a linear/non-linear function that associates each current value to its past values, predict the value of a record at a specific time, and report as anomalies those records whose prediction error falls outside a threshold. Stochastic modeling techniques, such as Moving Average (MA) [30], Autoregressive Integrated Moving Average (ARIMA) [31], and Holt-Winters (HW) [32] use statistical measures to calculate the correlation between the data records. These techniques assume that the time series is linear and follows a known statistical distribution, which make them inapplicable to many practical problems [33]. Machine learning modeling techniques support non-linear modeling, with no assumption about the distribution of the data [33]. Examples are multi layer perceptrons (MLPs) [34], long short term memory (LSTM) [9], and hierarchical temporal memory (HTM) [11]. Some of these techniques can model multivariate time-series. However, they produce complex equations, which are not human interpretable.

Existing techniques for anomalous sequence detection split the data into multiple subsequences, typically based on a fixed-size window [35] or an exhaustive brute-force approach [36]. Clustering-based anomalous sequence detection techniques extract subsequence features, such as trend and seasonality, and group the subsequences based on the similarities between their features. An anomalous subsequence is detected as the one that is distantly positioned within a cluster or is positioned in the smallest cluster. These approaches only detect anomalous sequences without determining the records and attributes that are the major causes of invalidity in each subsequence. Autoencoder-based techniques (1) take subsequences as input, (2) use an autoencoder network to reconstruct the subsequences, (3) assign invalidity scores based on the reconstruction errors to the subsequences, and (4) detect as anomalous those subsequences whose scores are greater than a threshold. These techniques can learn complex non-linear associations among the attributes in the time series but are not able to model the temporal dependencies among the records in the input subsequence. An LSTM-autoencoder extends an autoencoder for time series data, and captures long-term temporal associations among data records in the form of complex equations that are not human interpretable. Our work aims to fill this gap by illustrating the cause of anomaly to the domain experts.

Approach

Figure 1 shows an overview of our approach. The input is in the form of data records and the output consists of a report showing subsequences of suspicious records accompanied with an explanation of the violated constraints. There are five components, namely, data preparation, constraint discovery, anomaly detection, anomaly interpretation, and anomaly inspection. These components form the basis of IDEAL [19] and are briefly described in the following paragraphs. “Extension to data preparation” and “Evaluation” describe how the data preparation and anomaly interpretation components are extended in this paper.

Data preparation This component prepares the data by transforming raw data into a form suitable for analysis. We used the one-hot encoding [37] method for preprocessing categorical attributes and the normalization [38] method for numeric attributes. Moreover, we proposed a systematic reshaping approach that uses autocorrelation [39] of the time-series attributes to enable the LSTM-autoencoder network discover dependencies between highly correlated records. Note that, this step must be extended to handle COVID-19 data. The extensions are described in “Extension to data preparation”.

Constraint discovery IDEAL uses an LSTM-autoencoder, which is a sequence-to-sequence modeling technique [40] used to learn time series dependencies. An LSTM-autoencoder can discover constraints involving long-term non-linear associations among multivariate time-series data records and attributes. The input and output to this network are fixed-size time series matrices. The network is composed of two hidden LSTM layers. The first LSTM layer functions as an encoder that investigates the dependencies from the input sequence and produces a complex hidden context. The second LSTM layer functions as a decoder that reconstructs the input sequence, based on the learned complex context and the previous output state. The difference between the original input and the reconstructed input is termed as the reconstruction error. Appendix A describes the architecture of the LSTM-autoencoder network.

The LSTM-autoencoder is an unsupervised technique that can potentially learn incorrect constraints from invalid data and generate false alarms. IDEAL uses an interactive learning approach that takes the expert’s feedback through the anomaly inspection component to retrain the LSTM-Autoencoder model and improve its accuracy.

Anomaly detection This component detects suspicious subsequences and records that do not conform to the constraints represented by the trained model. Subsequence and records are assigned suspiciousness scores (s-scores), which are calculated based on the network reconstruction error and the record labels. The record label indicates the validity level of the record. If we start with an unlabeled data set, the labels of all records are 0. The record label changes as we incorporate domain expert feedback in the subsequent iterations. Subsequences and records whose scores are greater than a threshold are flagged as suspicious. Using record labels in the definition of s-scores ensures that no valid subsequences or records are reported as suspicious in the retraining phase, thereby minimizing false alarms.

Anomaly interpretation This component helps a domain expert interpret each suspicious subsequence by generating visualization plots of two types, namely, s-score per attribute and decision tree. The trained LSTM-autoencoder model calculates the s-score per attribute. The higher the value of s-score, the more likely is the attribute to contribute to the invalidity of the subsequence. For each subsequence, IDEAL plots the s-score values for all the attributes in the subsequence. Moreover, IDEAL uses a decision tree [41] based technique called random forest [42] classifier to determine the constraints that are violated by each suspicious

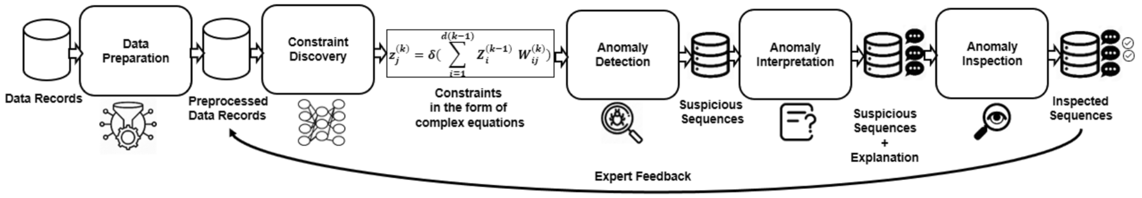

Data Records
Data Preparation
Preprocessed Data Records
Constraint Discovery
$z_j^{(k)} = \delta(\sum_{i=1}^{d(k-1)} z_i^{(k-1)} W_{ij}^{(k)})$
Constraints in the form of complex equations
Anomaly Detection
Suspicious Sequences
Anomaly Interpretation
Suspicious Sequences + Explanation
Anomaly Inspection
Inspected Sequences
Expert Feedback

Fig. 1 IDEAL overview [19]

subsequence. For each attribute of the subsequence, a set of time series features, such as Mean, Max, and Curvature are extracted using Tsfeatures [43] CRAN library. Next, decision trees are generated using these features. The decision trees represent a set of if-then-else decision rules, which describe the constraints that identify sequences as valid or invalid based on their feature values. Note that, our anomaly interpretation component is extended in "Evaluation".

Anomaly inspection This component takes domain expert feedback through a web-based user interface that uses check boxes for the expert to flag as faulty the subsequences that are actually anomalous. The feedback is incorporated to label the training data records as faulty or valid. The accuracy of constraint discovery is improved by adding the record label with four possible values (1: faulty, 0.5: suspicious, 0: unknown, and −1: valid) as a new attribute to the training data set. This label is updated using domain expert feedback in every interaction. We redefine the reconstruction error of LSTM-autoencoder based on the labels to minimize false alarms. The network is trained to minimize both the difference between the time series and its reconstruction, and the difference between the record labels in a time series and the labels predicted by the network.

Extension to Data Preparation

Analyzing COVID-19 time series data requires that the data be converted to a form suitable for analysis. In COVID-19 data, there are multiple subsets of data; records in each subset are temporally dependent but they are unrelated to records in other subsets. Such data must be prepared before being fed as input to sequential learning models, including the LSTM-based model used in this study, which assumes that all data records in an input sequence are temporally dependent and are describing behaviors of the same object over time.

A time series T is a sequence of d-dimensional records [44] described using the vector $T = \langle R_0, \dots, R_{t-1} \rangle$, where $R_i = (a_i^0, \dots, a_i^{d-1})$ is a record at time i, for $0 \le i \le t-1$ and a_i^j is the jth attribute of the ith record. A time series can be univariate ($d = 1$) or multivariate ($d > 1$) [45]. A univariate time series has one time-dependent attribute. For example, a univariate time series can consist of daily COVID-19 cases recorded sequentially over 24-h increments. A multivariate time series is used to simultaneously capture the dynamic nature of multiple attributes. For example, a multivariate time series from a health data store can consist of multiple laboratory results of patients over time.

Reshaping is an essential data preparation step for sequential learning models [46, 47]. This method reshapes the data to base the model computations at a time step t on a specified number of previous time steps. The number of previous time steps is known as window size.

Existing reshaping techniques use a single-level windowing approach, which assumes that all data records in a data set are temporally dependent and are describing behaviors of a single object over time. For example, all the traffic data in the Yahoo Benchmark data store [20] are records related to a single server. The NASA Shuttle data set [21] contains records of a single shuttle over time. However, real-world data sets including the ones used in this study typically contain records of multiple objects over time. For example, a COVID-19 tracking data set can store case records of multiple states over time in the US. A medical data set may contain clinical records for multiple patients over time. Each object (i.e., state in the COVID-19 data set and patient in the medical data set) has a unique id, which distinguishes the records concerning that object from the other records in the data set. Single-level reshaping techniques cannot be used to split the data records into multiple subsequences in such data sets. These techniques may generate subsequences with temporally unrelated records, which can result in generating false alarms. For example, Table 1 shows a portion of records (i.e., for Patient_ID = 1001 and Patient_ID = 1005) in a medical data set that stores patient weights over time.

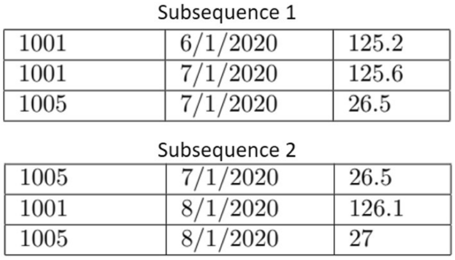
Splitting this data set through single-level reshaping using window size equal to three with one record overlap results in two subsequences (Fig. 2), which contain unrelated records.

However, the correct windowing must only contain temporally-related records of a single object (Fig. 3). An anomaly detection technique may incorrectly detect subsequence 1 and 2 in Fig. 2 as anomalous (i.e., two false positives) because of the sudden changes in the Weight values. Figure 3 shows the correct reshaping for our example.

In this work, we propose a two-level reshaping technique to address the above-mentioned issue. This technique (A) groups the time-series data based on domain-dependent grouping attributes, and (B) splits the data records in each group using our systematic autocorrelation-based reshaping [19] approach.

Table 1 Patient weights sequence

Patient_ID	Timestamp	Weight
1001	6/1/2020	125.2
1001	7/1/2020	125.6
1005	7/1/2020	26.5
1001	8/1/2020	126.1
1005	8/1/2020	27



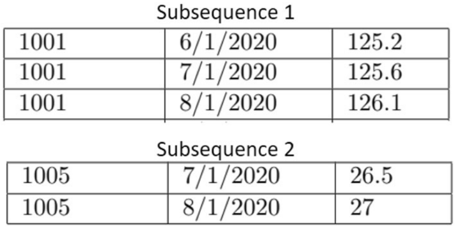
Subsequence 1

1001	6/1/2020	125.2
1001	7/1/2020	125.6
1005	7/1/2020	26.5

Subsequence 2

1005	7/1/2020	26.5
1001	8/1/2020	126.1
1005	8/1/2020	27

Fig. 2 Incorrect reshaping of patient records into multiple subsequences



Subsequence 1

1001	6/1/2020	125.2
1001	7/1/2020	125.6
1001	8/1/2020	126.1

Subsequence 2

1005	7/1/2020	26.5
1005	8/1/2020	27

Fig. 3 Correct reshaping of patient records into multiple subsequences

Grouping Data Records

We split the data records into multiple temporally dependent groups. For this purpose, we (1) identify grouping attributes and their hierarchy, (2) concatenate the non-first-level grouping attributes into a new attribute, (3) pivot the new attribute into multiple temporal attributes, and (4) use the first-level grouping attribute to split the data records into multiple temporally-dependent subsequences.

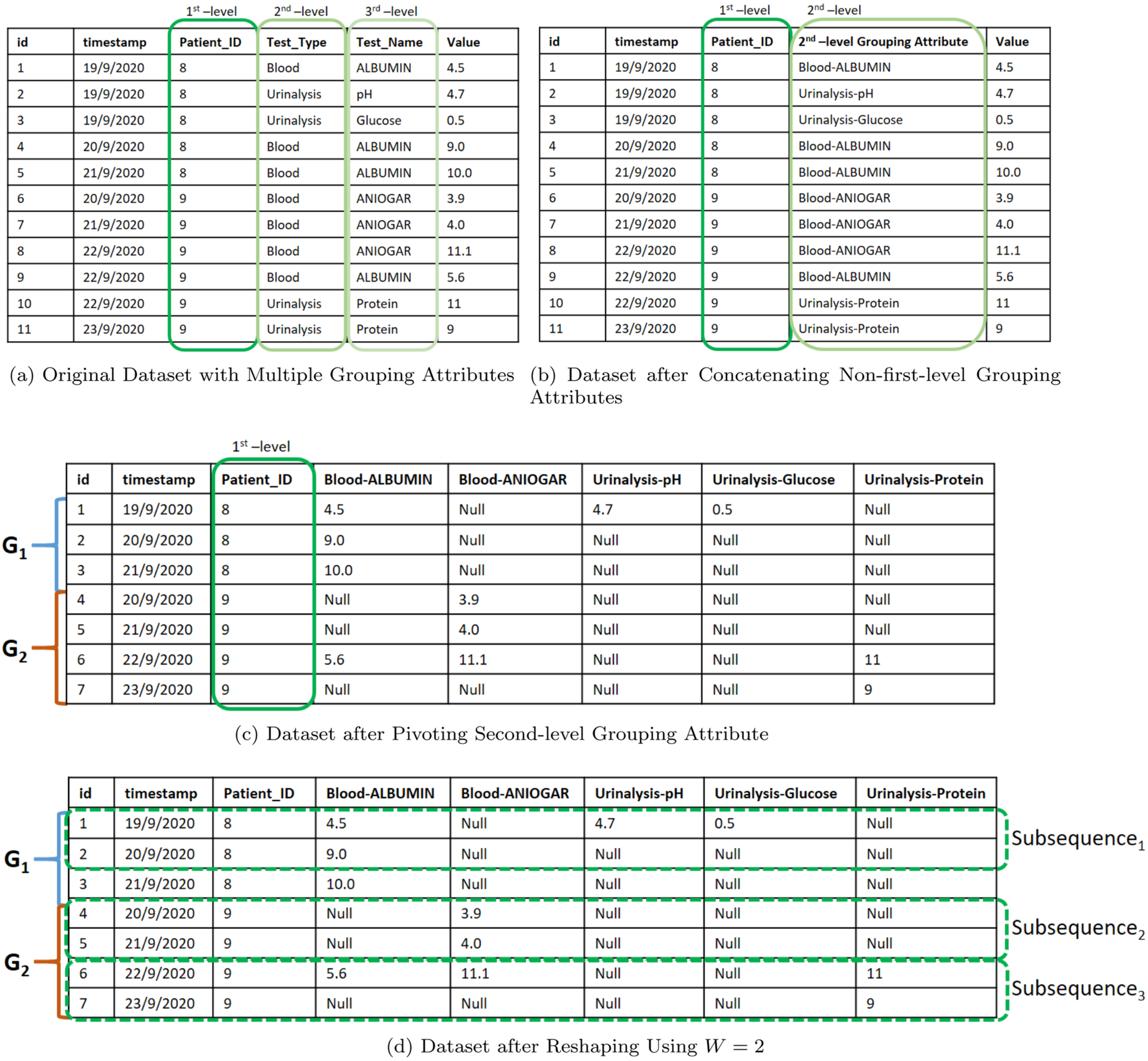
1. Identify grouping attributes. A grouping attribute is a categorical column by which we can group the data set records into multiple temporally-dependent subsequences. A data set may have one or more grouping attributes, which are domain-dependent. Figure 4a shows an example of a medical data set of laboratory results for multiple patients over time. This data set contains three levels of grouping attributes to describe the data records. The first-level grouping attribute (i.e., Patient_ID) indicates the objects in a data set, each of which is represented by a unique Id. The second- to *h*-level grouping attributes indicate features about those objects. For example, in Fig. 4a, the second-level grouping attribute (i.e., Test_Type) represents the type of laboratory test. Each patient can receive multiple types of test. The third-level grouping attribute (i.e., Test_Name) is the name of the laboratory test performed on the patients. Each Test_Type includes multiple Test_Name. We identify the domain-dependent grouping attributes with the help from domain experts. In the future, we will use statistical autocorrelation-based [48] techniques to automatically identify the grouping attributes and their hierarchy from an input data set.
2. Concatenate non-first-level grouping attributes. Our approach converts all the non-first-level grouping attributes into a single data set column to reduce the complexity (i.e., dimensionality) of the problem. We call this new column the second-level grouping attribute. Figure 4b shows the new generated column from all the non-first-level grouping attributes.

   At this step, we can use the first- and second-level attributes to group the data into multiple temporally dependent subsequences. However, using the resulting subsequences generated by this approach, time-series analysis techniques will not preserve the associations among the values of second-level attribute if any. For example, if there are associations among "Blood-Sugar" and "Blood-Pressure" of a patient, grouping at this stage would not preserve this association. To address this issue, our approach uses another step for pivoting the second-level attribute into multiple temporal attributes based on the attribute values.
3. Pivot second-level grouping attribute. A pivoting query [49] converts all unique rows of an attribute into separate columns of their own, each of which contains a value specified as an input to the query. IDEAL pivots the second-level grouping attribute to generate multiple temporal attributes. The objective is to preserve the associations among grouping attribute values. Pivoting results in a smaller number of records in comparison to the original data set.

   Figure 4c shows how the pivoting process works in the example data set. The second-level grouping attribute (i.e., a concatenation of Test_Type and Test_Name) is converted into five new attributes, each of which contains the corresponding test result stored in the value attribute. If a patient has not received a specific test at a specific time, the value of that test is set to Null.
4. Group records by first-level grouping attribute. We use the first-level grouping attribute to categorize the data records in a sequence data set into multiple groups G_i, $1 \le i \le m$, where m is the number of the distinct values of that attribute. Figure 4c shows the two groups of temporally-dependent records (i.e., G_1 and G_2) generated for the example data set.

Autocorrelation-Based Reshaping of Groups

For each group, IDEAL uses a systematic reshaping approach that we proposed in an earlier work [19] to split the data records in that group. This approach is based on the autocorrelation of the time series attributes to enable the LSTM-autoencoder


1st –level
2nd –level
3rd –level

id	timestamp	Patient_ID	Test_Type	Test_Name	Value
1	19/9/2020	8	Blood	ALBUMIN	4.5
2	19/9/2020	8	Urinalysis	pH	4.7
3	19/9/2020	8	Urinalysis	Glucose	0.5
4	20/9/2020	8	Blood	ALBUMIN	9.0
5	21/9/2020	8	Blood	ALBUMIN	10.0
6	20/9/2020	9	Blood	ANIOGAR	3.9
7	21/9/2020	9	Blood	ANIOGAR	4.0
8	22/9/2020	9	Blood	ANIOGAR	11.1
9	22/9/2020	9	Blood	ALBUMIN	5.6
10	22/9/2020	9	Urinalysis	Protein	11
11	23/9/2020	9	Urinalysis	Protein	9

(a) Original Dataset with Multiple Grouping Attributes

id	timestamp	Patient_ID	2nd –level Grouping Attribute	Value
1	19/9/2020	8	Blood-ALBUMIN	4.5
2	19/9/2020	8	Urinalysis-pH	4.7
3	19/9/2020	8	Urinalysis-Glucose	0.5
4	20/9/2020	8	Blood-ALBUMIN	9.0
5	21/9/2020	8	Blood-ALBUMIN	10.0
6	20/9/2020	9	Blood-ANIOGAR	3.9
7	21/9/2020	9	Blood-ANIOGAR	4.0
8	22/9/2020	9	Blood-ANIOGAR	11.1
9	22/9/2020	9	Blood-ALBUMIN	5.6
10	22/9/2020	9	Urinalysis-Protein	11
11	23/9/2020	9	Urinalysis-Protein	9

(b) Dataset after Concatenating Non-first-level Grouping Attributes

id	timestamp	Patient_ID	Blood-ALBUMIN	Blood-ANIOGAR	Urinalysis-pH	Urinalysis-Glucose	Urinalysis-Protein
1	19/9/2020	8	4.5	Null	4.7	0.5	Null
2	20/9/2020	8	9.0	Null	Null	Null	Null
3	21/9/2020	8	10.0	Null	Null	Null	Null
4	20/9/2020	9	Null	3.9	Null	Null	Null
5	21/9/2020	9	Null	4.0	Null	Null	Null
6	22/9/2020	9	5.6	11.1	Null	Null	11
7	23/9/2020	9	Null	Null	Null	Null	9

(c) Dataset after Pivoting Second-level Grouping Attribute

id	timestamp	Patient_ID	Blood-ALBUMIN	Blood-ANIOGAR	Urinalysis-pH	Urinalysis-Glucose	Urinalysis-Protein
1	19/9/2020	8	4.5	Null	4.7	0.5	Null
2	20/9/2020	8	9.0	Null	Null	Null	Null
3	21/9/2020	8	10.0	Null	Null	Null	Null
4	20/9/2020	9	Null	3.9	Null	Null	Null
5	21/9/2020	9	Null	4.0	Null	Null	Null
6	22/9/2020	9	5.6	11.1	Null	Null	11
7	23/9/2020	9	Null	Null	Null	Null	9

(d) Dataset after Reshaping Using $W = 2$

Fig. 4 Splitting a data set with multiple grouping attributes into multiple temporally-dependent subsequences

network discover dependencies between the records that are highly correlated. The input size is adjusted based on how far the records in a group are related to their past values. By feeding the LSTM-autoencoder network with highly correlated records, this reshaping approach prevents the network from incorrectly discovering associations among non-correlated records. For each group G_i, IDEAL uses the autocorrelation-based approach to identify the window size w_i for that group. Autocorrelation is defined as the correlation of sequence data records with the records in the previous time steps, called lags [39]. An Autocorrelation Function (ACF [48]) at lag k for an attribute identifies to what extend the attribute is correlated to its kth past value. IDEAL calculates ACF to identify the lags at which the attribute values are highly correlated to set the window size. As the LSTM-autoencoder window size must be similar for all the data records in a data set, IDEAL sets the final value of window size W to the smallest value of window sizes calculated for the groups (Eq. 1). Finally, our approach reshapes the data records in groups based on the value of W.

$$W = \text{Min}(w_i),\ 1 \leq i \leq m, \tag{1}$$

where m is the number of distinct values of the grouping attribute. Figure 4d shows how IDEAL splits the records in each group into multiple subsequences for $W = 2$.

Extension to Anomaly Interpretation

IDEAL uses an additional data visualization plot to explain the level of abnormality of suspicious subsequences. This plot visualizes attribute values for multiple groups (e.g., state in COVID-19 data set and patient in medical data set) over time. The visualization plot uses color-coded diagrams for every group to help a domain expert compare a suspicious subsequence with other subsequences from other groups in the data set.

For each suspicious subsequence, IDEAL uses *s*-score per attribute values [19] to select the attribute with the highest suspiciousness score. Next, IDEAL plots values of that attribute for all the groups (i.e., G_i, $1 \leq i \leq m$) over time. The attribute values of the suspicious subsequence are represented by red points. Figures 5 and 6 shows the visualization plots generated for a suspicious subsequence detected from the laboratory results in the Anschutz medical data. Figure 5 shows that e_TOTGLOB attribute (i.e., total serum globulin) is the major cause of invalidity (i.e., attribute with the highest *s*-score value) in this subsequence. The data visualization plot in Fig. 6 shows the values of the e_TOTGLOB attribute over time for this subsequence (in red) as well as other subsequences (in colors other than red) in the same data set. We can visually observe from this figure how the values of e_TOTGLOB for this patient deviate from those of the majority of the patients in the data set. As the value of this attribute is elevated in certain immunological diseases, this deviation can be caused by an immunological disease of the patient.

Evaluation

We evaluated the anomaly detection effectiveness of IDEAL using COVID-19 records from Johns Hopkins (JH) [23], New York Times (NT) [24], and Tracking project (T) [25] repositories. These publicly available data sets are updated daily and contain county- or state-level COVID-19 attributes. Wissel et al. [50] compared these data sets based on different factors, such as their data sources, collected attributes, region granularity, and frequency of updates. We used nine-month data from March 5th to November 11th, 2020 to evaluate one execution of IDEAL, which is an execution without the feedback loop. Moreover, we used four health data sets from the University of Colorado Anschutz medical campus [26] to evaluate the anomaly detection effectiveness of IDEAL. We used records of COVID-19-positive patients to evaluate one execution of IDEAL.

Current knowledge about the COVID-19 data attributes, pattern of spread, and distribution is insufficient as this is an unprecedented pandemic. We used two evaluation approaches to validate the suspicious subsequences detected from the COVID-19 data in the domain knowledge absence; these are (1) comparing suspicious subsequences detected by IDEAL from one data source to those from other independent sources that are recording values of the same records and attributes and (2) comparing the suspicious subsequences

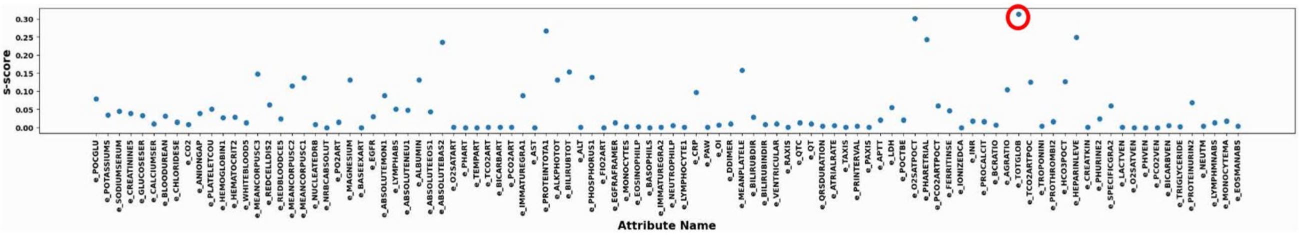


Fig. 5 *s*-score per attribute plot for suspicious subsequence detected from Lab results from Anschutz Medical Campus

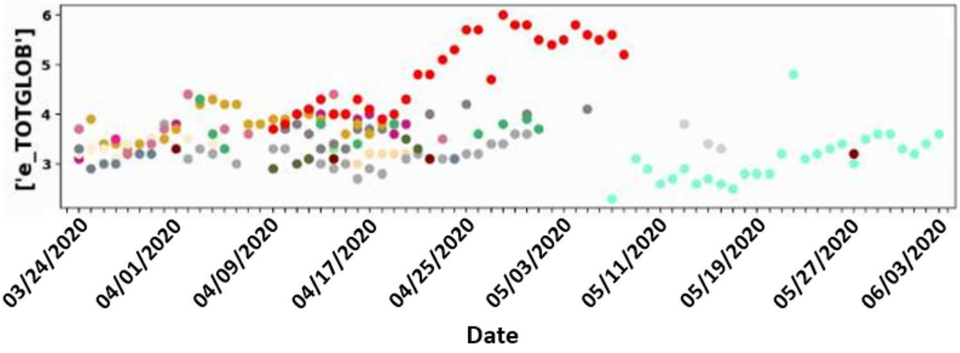


Fig. 6 Data visualization plot for e_TOTGLOB attribute

detected by IDEAL from a homogeneous population with the other subsequences in that population.

Comparing Suspicious Subsequences from Different Sources (Johns Hopkins (JH), New York Times (NT), and Tracking Project (T))

The objective is to identify whether or not a suspicious subsequence is actually anomalous by comparing the suspicious subsequence from a data source with its equivalent subsequences from other independent COVID-19 data repositories. Two subsequences are equivalent if they contain records of same object (i.e., same grouping attribute value) and are observed during the same time period. For example, data records of the Alabama state collected from three sources of data during March to April 2020 form three equivalent subsequences. We formalized possible observations on equivalent subsequences to validate a suspicious subsequence based on whether (1) the same subsequence is detected by IDEAL as suspicious from all available sources of data or (2) the subsequence is detected as anomalous only in some of the available sources. We decided on whether or not each suspicious subsequence is actually anomalous based on a distance measure (i.e. mean square error (MSE)) between the attribute values of the suspicious subsequence detected from a source with those of equivalent subsequences collected from the other data sources.

For each suspicious subsequence s_i detected from the ith source in a set of sources D (where $|D| = n$), we calculated the mean square error value between s_i and all its equivalent subsequences s_j from the other sources ($1 \leq j \leq n$ and $j \neq i$). s_j can be either an undetected or a suspicious subsequence:

$$\mathrm{MSE}(s_i, s_j) = \frac{1}{w}\sum_{k=1}^{w}(\mathrm{Normalized}(A_i^f) - \mathrm{Normalized}(A_j^f))^2, \tag{2}$$

where A_i^f attribute is the major cause of invalidity in s_i, A_j^f is its equivalent attribute in s_j, and w is the window size. The following four cases describe how we validated a suspicious subsequence s_i based on the MSE value.

(A) If attribute values of equivalent subsequences to s_i collected from all other sources of data are close to values of s_i, then all of those subsequences are either abnormal but valid, or anomalous detected from the same source. An abnormal subsequence can indicate signals of a COVID-19 outbreak [5]. If $\forall j \in \{1, \dots, n\}, \mathrm{MSE}(s_i, s_j) \leq \mathrm{Threshold}$, then s_i and s_j are either

- Abnormal but valid, or
- Anomalous collected from the same source

(B) If attribute values of equivalent subsequences to s_i collected from all other sources of data are far from values of s_i but close to one another, then s_i is anomalous: If $\forall j, k \in \{1, \dots, n\}(j, k \neq i), \mathrm{MSE}(s_i, s_j) > \mathrm{Threshold}$ and $\mathrm{MSE}(s_j, s_k) \leq \mathrm{Threshold}$, then s_i is:

- Anomalous

(C) If attribute values of s_i are close to a subset of equivalent subsequences to s_i but far from another subset of equivalent subsequences, then all subsequences in the smaller subset are anomalous: If $\forall j \in D_1,\ k \in D_2\ (D_1 \cup D_2 = D,\ D_1 \cap D_2 = \emptyset,\ |D_1| < |D_2|)$, $\mathrm{MSE}(s_i, s_j) \leq \mathrm{Threshold}$ and $\mathrm{MSE}(s_i, s_k) > \mathrm{Threshold}$, then:

- s_i and s_j are anomalous, and
- s_k is valid

(D) If attribute values of s_i are close to a subset of equivalent subsequences to s_i but far from another subset of equivalent subsequences and the two subsets are of the same size, then all subsequences in both subsets are abnormal and need more investigations by a domain expert: If $\forall j \in D_1,\ k \in D_2\ (D_1 \cup D_2 = D,\ D_1 \cap D_2 = \emptyset,\ |D_1| = |D_2|)$, $\mathrm{MSE}(s_i, s_j) \leq \mathrm{Threshold}$ and $\mathrm{MSE}(s_i, s_k) > \mathrm{Threshold}$, then:

- s_i, s_j, and s_k are abnormal and need more investigation

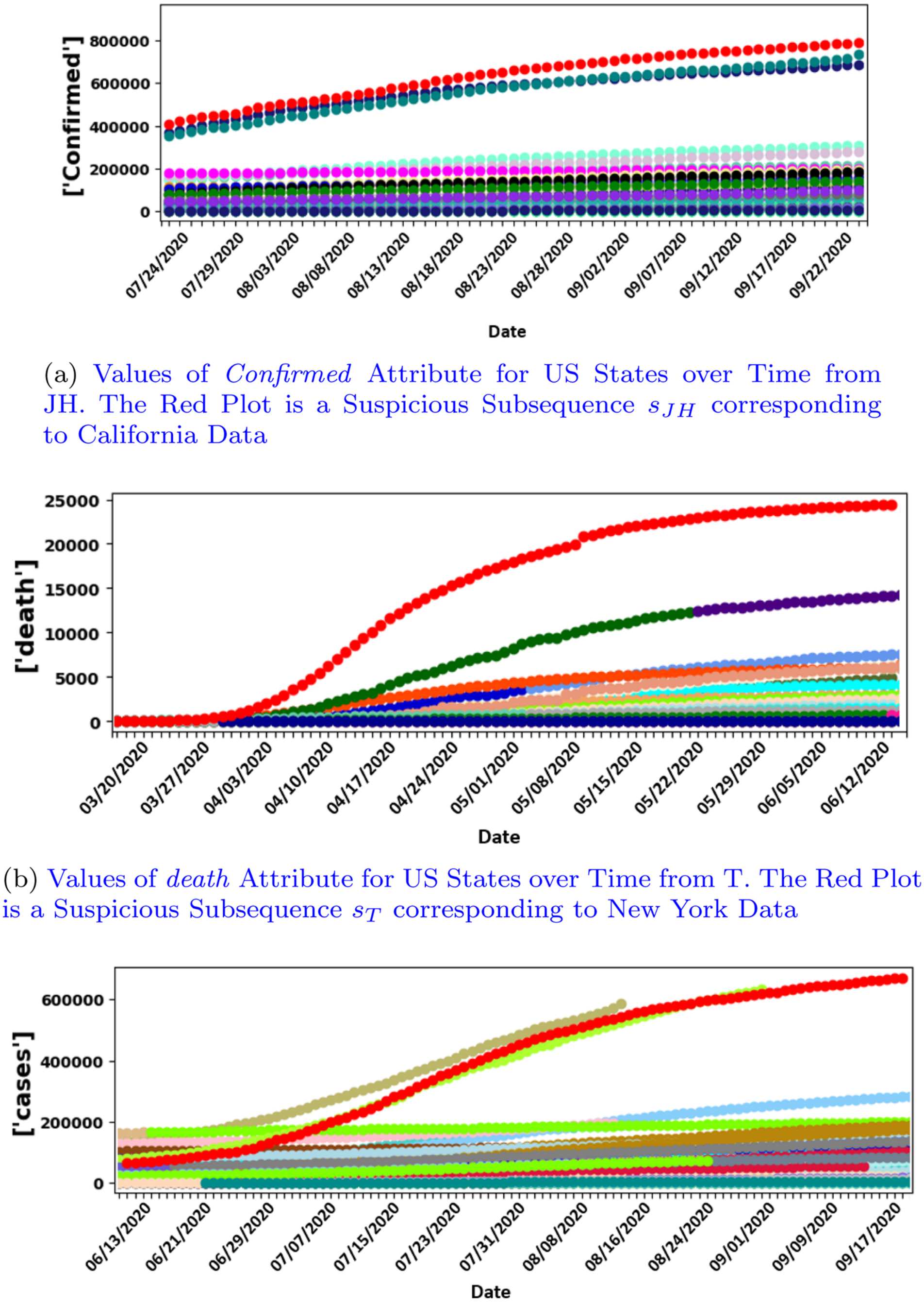
We ran two experiments. Table 2 shows the attributes from the COVID-19 data sets used for each experiment. Figure 7 shows data visualization plots for suspicious subsequences detected by IDEAL from data sets of the first experiment. In this figure, each color represents data of a state. There are 50 plots for the 50 states of the US. The red plot represents the data of the suspicious subsequence. We used Fig. 8 to validate the suspicious subsequences by comparing the attribute values of the suspicious subsequences with those of their equivalent subsequences from the first experiment data sets. These attributes are major causes of invalidity in each suspicious subsequence. In this experiment, the threshold T is set at 0.03 based on our observations of the values of MSE in these data sets.

Figure 7a: A suspicious subsequence s_{JH} was detected from JH in California data. An equivalent subsequence s_{T} was detected from T. The Confirmed attribute was the major cause of invalidity in these subsequences. The data visualization plot in Fig. 7a shows how the Confirmed attribute values of the suspicious subsequence from California data in JH (red points) deviate from other subsequences from other states in the same source (i.e., JH). The constraint violations reported by the decision trees for this suspicious subsequence were over the Minimum and Mean features of the subsequence. In Fig. 8a, $\mathrm{MSE}(s_{\mathrm{JH}}, s_{\mathrm{T}}) < T$,

Table 2 Data sets

Experiment ID	Data sets (sources)	Attributes
1	State-level data from JH, NT, and T	Confirmed Cases (i.e., JH.Confirmed, NT.cases, T.positive) and Deaths
2	State-level data from JH and T	Recovered and Hospitalization_Rate

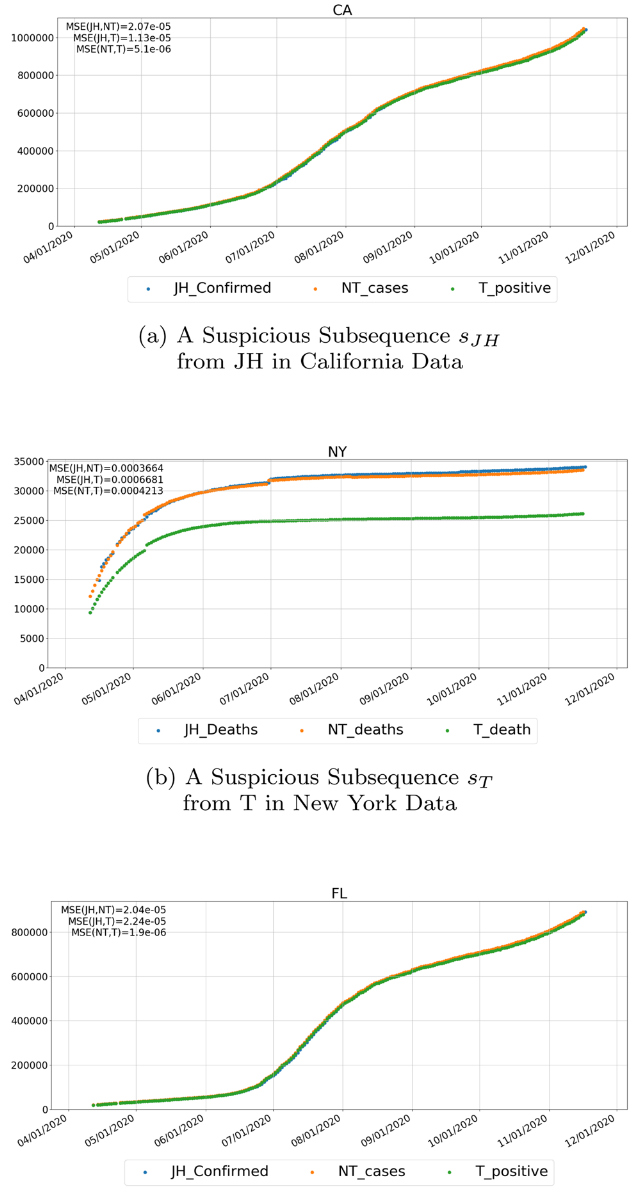
Fig. 7 Visualization plots for suspicious subsequences detected from data sets of experiment 1



(a) Values of *Confirmed* Attribute for US States over Time from JH. The Red Plot is a Suspicious Subsequence s_{JH} corresponding to California Data

(b) Values of *death* Attribute for US States over Time from T. The Red Plot is a Suspicious Subsequence s_T corresponding to New York Data

(c) Values of *cases* Attribute for US States over Time from NT. The Red Plot is a Suspicious Subsequence s_{NT} corresponding to Florida Data



(a) A Suspicious Subsequence s_{JH} from JH in California Data

(b) A Suspicious Subsequence s_T from T in New York Data

(c) A Suspicious Subsequence s_T from NT in Florida Data

Fig. 8 Actual attribute values in suspicious subsequences detected from data sets of experiment 1

MSE($s_{\mathrm{JH}}, s_{\mathrm{NT}}$) $< T$, and MSE($s_{\mathrm{T}}, s_{\mathrm{NT}}$) $< T$. This result indicates that in California, the numbers of confirmed cases over time reported by all three data sources for this suspicious subsequence are close to each other. Based on the case A, s_{JH}, s_{NT}, and s_{T} are either (1) abnormal but valid or (2) anomalous data that have been obtained from the same source.

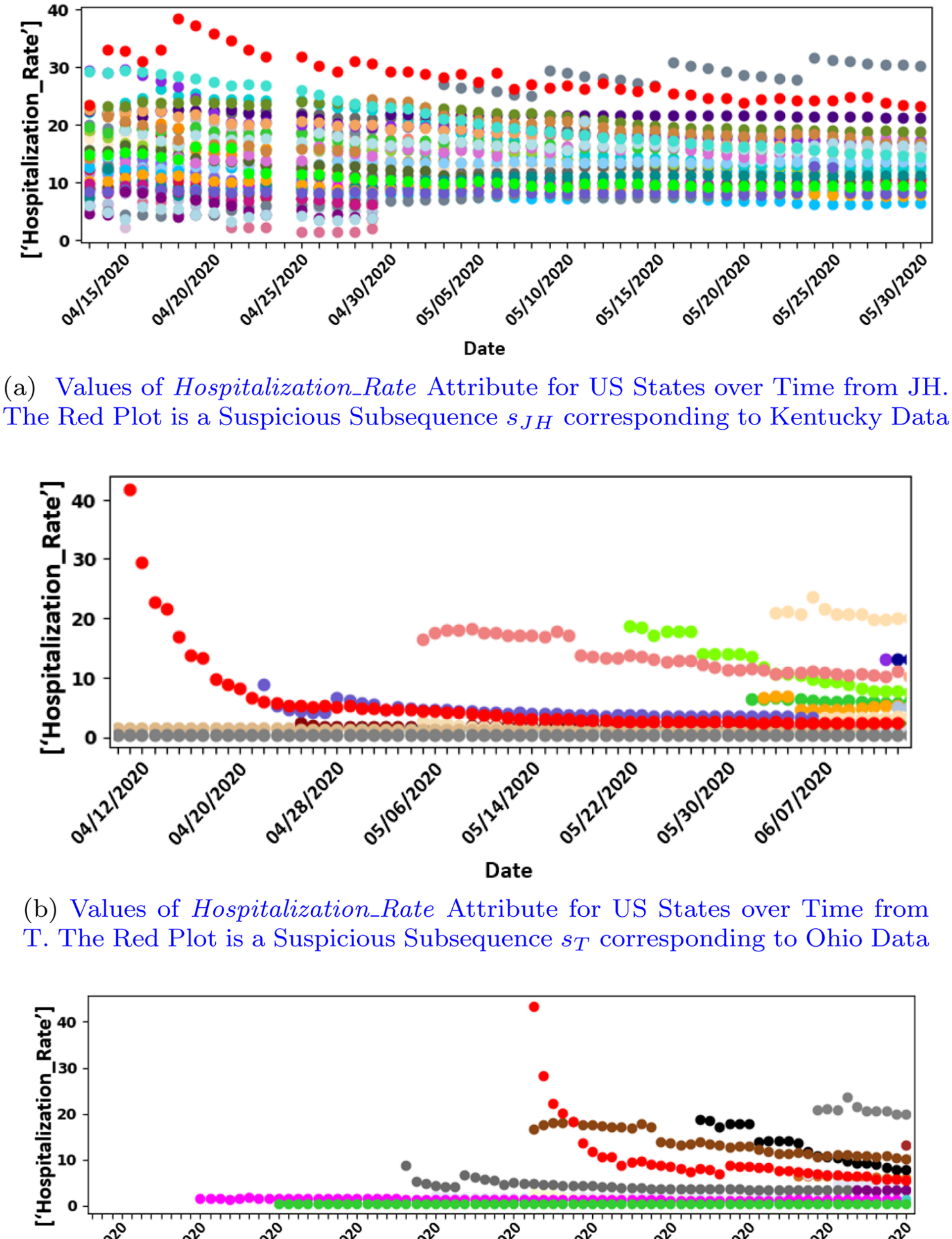
Figure 7b: A suspicious subsequence s_{T} was detected from T in New York data. Equivalent subsequences s_{JH} and s_{NT} were also detected from JH and NT. The Deaths attribute was the major cause of invalidity in these subsequences. The data visualization plot in Fig. 7b shows how the *Deaths* attribute values of the suspicious subsequence from New York data in T (red points) deviate from other subsequences from other states in the same source (i.e., T). The constraint violations reported by the decision trees for this suspicious subsequence were over the *Linearity* (i.e., strength of linearity, which is the sum of squared residuals of time-series from a linear autoregression) and *Burstiness* (i.e., ratio between the variance and the mean (Fano Factor) of time series) features of the subsequence. In Fig. 8b, MSE($s_{\mathrm{JH}}, s_{\mathrm{T}}$) $> T$, MSE($s_{\mathrm{NT}}, s_{\mathrm{T}}$) $> T$, and MSE($s_{\mathrm{JH}}, s_{\mathrm{NT}}$) $< T$. This result indicates that the number of death cases in the New York state collected by the data source T was considerably less than that of the other two sources of data. Based on case B, s_{T} is anomalous and s_{JH}, s_{NT} are valid.

Figure 7c: A suspicious subsequence s_{NT} was detected only from NT in Florida data. The Confirmed attribute was the major cause of invalidity in this subsequence. The data visualization plot in Fig. 7c shows how the Confirmed attribute values of the suspicious subsequence from Florida data in NT (red points) deviate from other subsequences from other states in the same source (i.e., NT). The constraint violations reported by the decision trees for this suspicious subsequence were over the Mean and Curvature (i.e., strength of curvature, which is the amount by which a time series curve deviates from being a straight line and calculated based on the coefficients of an orthogonal quadratic regression) features of the subsequence. In this figure, MSE($s_{\mathrm{JH}}, s_{\mathrm{T}}$) $< T$, MSE($s_{\mathrm{JH}}, s_{\mathrm{NT}}$) $< T$, and MSE($s_{\mathrm{T}}, s_{\mathrm{NT}}$) $< T$. This result indicates that in Florida, the numbers of confirmed cases over time reported by all three data sources for this suspicious subsequence are close to each other. Based on case A, $s_{\mathrm{JH}}, s_{\mathrm{NT}}, s_{\mathrm{T}}$ are either valid or anomalous collected from the same source.

Figure 9 shows data visualization plots for suspicious subsequences detected by IDEAL from data sets of the second experiment. We used Fig. 10 to validate the suspicious subsequences by observing the actual values of attributes for the suspicious subsequences and their equivalent subsequences from the second experiment data sets. These attributes are major causes of invalidity in each suspicious subsequence. In this experiment threshold T is set at 0.0004 based on our observations on the values of MSE in these data sets.

Figure 9a: A suspicious subsequence s_{JH} was detected only from JH in Kentucky data. The Hospitalization_Rate attribute was the major cause of invalidity in this subsequence. The data visualization plot in Fig. 9a shows how the *Hospitalization_Rate* attribute values of the suspicious subsequence (red points) from Kentucky data in JH deviate from other subsequences from other states in the same source (i.e., JH). The constraint violations reported by the decision trees for this suspicious subsequence were over the Mean, Maximum, and Vchange (i.e., maximum difference in variance between consecutive blocks in time series)

Fig. 9 Actual attribute values in suspicious subsequences detected from data sets of experiment 2



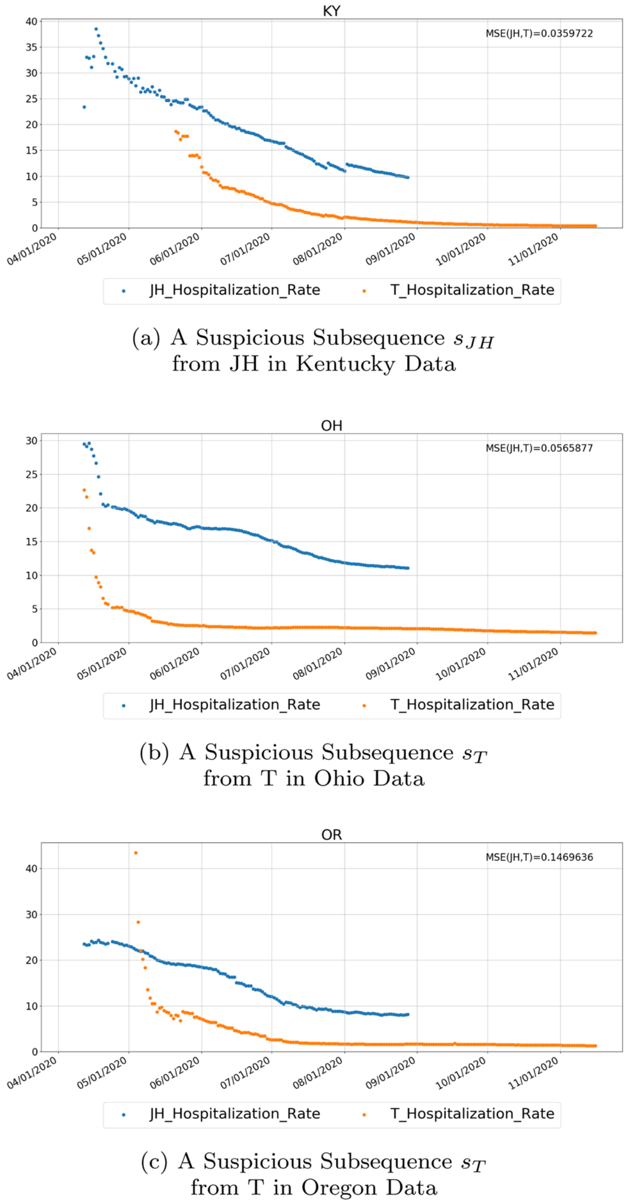
(a) Values of *Hospitalization_Rate* Attribute for US States over Time from JH. The Red Plot is a Suspicious Subsequence s_{JH} corresponding to Kentucky Data

(b) Values of *Hospitalization_Rate* Attribute for US States over Time from T. The Red Plot is a Suspicious Subsequence s_T corresponding to Ohio Data

(c) Values of *Hospitalization_Rate* Attribute for US States over Time from T. The Red Plot is a Suspicious Subsequence s_T corresponding to Oregon Data

features of the subsequence. In Fig. 10a, MSE($s_{\mathrm{JH}}, s_{\mathrm{T}}$) > T. This result indicates that the hospitalization rates reported by the two sources of data for the Kentucky state were considerably distinct. Based on case D, s_{JH} and s_{T} are abnormal subsequences that need more investigation.

Figure 9b: A suspicious subsequence s_{T} was detected only from T in Ohio data. The Hospitalization_Rate attribute was the major cause of invalidity in this subsequence. The data visualization plot in Fig. 9b shows how the Hospitalization_Rate attribute values of the suspicious subsequence from Ohio data in T (red points) deviate from other subsequences from other states in the same source. The constraint violations reported by the decision trees for this suspicious subsequence were over the Mean, Curvature (i.e., strength of curvature, which is the amount by which a time series curve deviates from being a straight line and calculated based on



(a) A Suspicious Subsequence s_{JH} from JH in Kentucky Data

(b) A Suspicious Subsequence s_T from T in Ohio Data

(c) A Suspicious Subsequence s_T from T in Oregon Data

Fig. 10 Actual attribute values in suspicious subsequences detected from data sets of experiment 2

the coefficients of an orthogonal quadratic regression), and Highlowmu (i.e., ratio between the means of data that is below and upper the global mean of time series) features of the subsequence. In Fig. 10b, MSE$(s_{\mathrm{JH}}, s_{\mathrm{T}}) > T$. This result indicates that the hospitalization rates reported by the two sources of data for the Ohio state were considerably distinct. Based on case D, s_{JH} and s_{T} are abnormal subsequences that need more investigation.

Figure 9c: A suspicious subsequence s_{T} was detected only from T in Oregon data. The Hospitalization_Rate attribute was the major cause of invalidity in this subsequence. The data visualization plot in Fig. 9c shows how the Hospitalization_Rate attribute values of the suspicious subsequence from Oregon data in T (red points) deviate from other subsequences from other states in the same source. The constraint violations reported by the decision trees for this suspicious subsequence were over the Variance and Burstiness (i.e., ratio between the variance and the mean (Fano Factor) of time series) features of the subsequence. In Fig. 10c, MSE$(s_{\mathrm{JH}}, s_{\mathrm{T}}) > T$. This result indicates that the hospitalization rates reported by the two sources of data for the Oregon state were considerably distinct. Based on case D, s_{JH} and s_{T} are abnormal subsequences that need more investigation.

Comparing Suspicious Subsequences from a Homogeneous Population

The data of the patients of a homogeneous population should relatively look similar. We used this idea as a relative goal to evaluate the COVID-19 data in the absence of a domain expert. For this purpose, we extracted data of four homogeneous populations from Anschutz medical data store (Table 3). These data sets are results of joins of multiple tables (i.e., Patient, Diagnosis, and Lab) in the Anschutz health data warehouse. We fed each population data as an input data set to the IDEAL tool to detect suspicious subsequences.

We validated the suspicious subsequences by visually observing the data visualization plots generated by IDEAL; we identified as actually abnormal (true positive) those suspicious subsequences of patients whose attribute values changing pattern over time are considerably different (i.e., visually observable) from other patients in their population. We identified as normal (false positive) those suspicious subsequences of patients whose attribute values changing pattern over time are not different (i.e., not visually observable) from other patients in their population.

Figure 11 shows a visualization plot generated by IDEAL for a suspicious subsequence detected by IDEAL from data set ID = 1. In this example, the e_Meancorpusc3 attribute is the major cause of suspiciousness of the subsequence. We can visually observe that the suspicious subsequence represented by red data points shows a considerable difference with other subsequences of the same population (i.e., COVID-positive with diabetes). As a result, this subsequence is a true positive. The constraint violations reported by the decision trees for this suspicious subsequence were over the Mean and Variance features of the subsequence.

Figure 12 shows a visualization plot generated by IDEAL for a suspicious subsequence detected by IDEAL from data set ID=3. In this example, the Phart attribute is the major cause of suspiciousness of the subsequence. We cannot visually observe a considerable difference between the suspicious subsequence represented by red

Table 3 Health data sets

Data set ID	Data set name	#records	#attributes
1	COVID-positive with diabetes	770	103
2	COVID-positive females over 60	1174	103
3	COVID-positive with hypertension	1270	103
4	COVID-positive males over 60	1839	103

data points and other subsequences of the same population (i.e., COVID-positive with hypertension). As a result, this subsequence is a false positive. The constraint violations reported by the decision trees for this suspicious subsequence were over the Maximum, Curvature (i.e., strength of curvature, which is the amount by which a time series curve deviates from being a straight line and calculated based on the coefficients of an orthogonal quadratic regression), and Linearity (i.e., strength of linearity, which is the sum of squared residuals of time series from a linear autoregression) features of the subsequence.

Table 4 shows number of true positives (TP), number of false positives (FP), Precision = $\frac{TP}{(TP+FP)}$ and total time (TT) it took to run the automated steps of IDEAL against each data set under test. As the data is unlabeled, we cannot calculate the recall metric, which is based on the number of false negatives. The number of true positives and false positives are calculated based on our observation on the data visualization plots. It took between 58 and 108 s to run IDEAL against the data sets. IDEAL could detect between 1 and 4 abnormal subsequences in these data sets. The precision was between 75 and 100%.

Conclusions

We extended our previous data quality test approach to address the problem of anomaly detection in data pertaining to COVID-19. We (1) proposed a two-level reshaping technique for data preparation, (2) added a data visualization plot for anomaly explanation, and (3) evaluated the approach against different COVID-19 data sets in the domain knowledge absence. We ran two experiments to validate the suspicious subsequences detected by IDEAL from Johns Hopkins, New York Times, and COVID-19 tracking project. We compared the attribute values of the suspicious subsequence detected from a source with those collected from other sources of data. IDEAL could find an anomalous subsequence in COVID-19 Tracking data set in the number of deaths in the New York state. IDEAL could find three abnormal subsequences in the three sources, which need more investigation by domain experts.

We also evaluated the anomaly detection effectiveness of IDEAL using four health data sets from Anschutz medical campus. We compared a suspicious supsequence with other subsequences in a homogeneous population. IDEAL could detect ten abnormal subsequences in these data sets.

In the future, we will evaluate the approach using other types of COVID-19 time series data. We plan to extend IDEAL to find anomalies in streaming COVID-19 data.

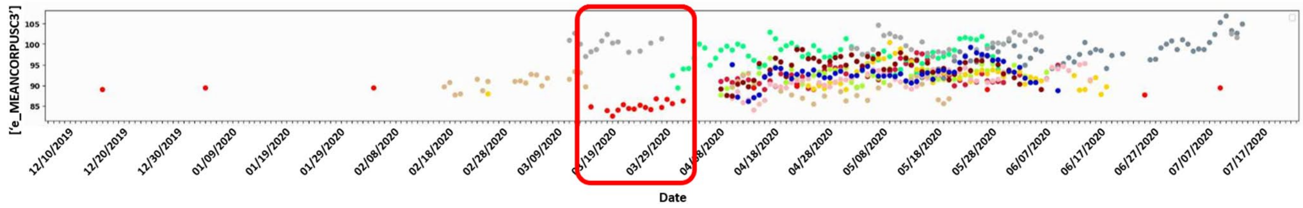


Fig. 11 Data visualization plot for a suspicious subsequence from COVID-positives with diabetes

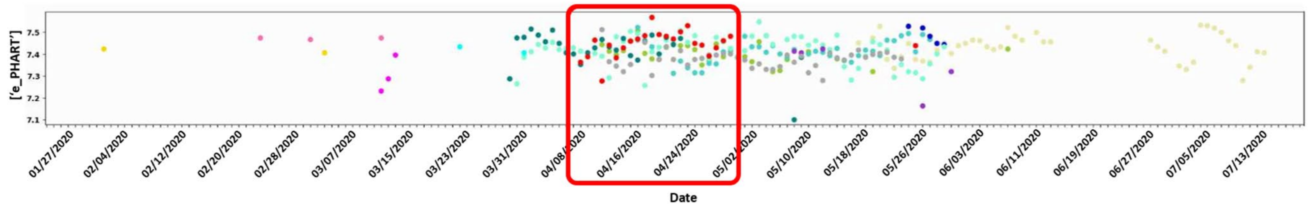


Fig. 12 Data visualization plot for a suspicious subsequence from COVID-positives with hypertension

Table 4 Results for health data sets

Data set ID	TP	FP	Precision	TT (s)
1	4	1	0.80	58
2	3	1	0.75	76
3	2	0	1.00	64
4	1	0	1.00	108

Appendices

An LSTM-Autoencoder

A Long Short Term Network (LSTM) [9] is a Recurrent Neural Network (RNN) [51] that contains loops in its structure to allow information to persist and make network learn sequential dependencies among data records [9]. An RNN can be represented as multiple copies of a neural network, each passing a value to its successor. The original RNNs can only learn short-term dependencies among data records using the recurrent feedback connections [45]. LSTMs extend RNNs using specialized gates and memory cells in their neuron structure to learn long-term dependencies. The computational units (neurons) of an LSTM are called memory cells. An LSTM has the ability to remove or add information to the memory cell state using gates. The gates are defined as weighted functions that govern information flow in the memory cells. The gates are composed of a sigmoid layer and a point-wise operation to optionally let information through. The sigmoid layer outputs a number between zero (to let nothing through) and one (to let everything through). There are three types of gates, namely, forget, input, and output.

- *Forget gate* Decides what information to discard from the memory cell. Equation 3 shows the mathematical representation of the forget gate.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f), \tag{3}$$

 where W_f is the connection weight between the inputs (h_{t-1} and x_t) and the sigmoid layer; b_f is the bias term and σ is the sigmoid activation function. In this gate, $f_t = 1$ means that completely keep the information and $f_t = 0$ means that completely get rid of the information.
- *Input gate* Decides which values to be used from the network input to update the memory state. Equation 4 shows the mathematical representation of the input gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{4}$$

 where C_t is the new memory cell state and C_{t-1} is the old cell state, which is multiplied by f_t to forget the information decided by the forget gate; $\tilde{C}_t$ is the new candidate value for the memory state, which is scaled by i_t as how much the gate decides to update the state value.
- *Output gate* Decides what to output based on the input and the memory state. Equation 5 shows the mathematical representation of the output gate. This gate pushes the cell state values between −1 and 1 by using a hyperbolic tangent function and multiplies it by the output of its sigmoid layer to decide which parts of the input and the cell state to output.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tan\mathrm{h}(C_t) \end{aligned} \tag{5}$$

An autoencoder is an unsupervised deep neural network that discovers constraints in the unlabeled input data. An autoencoder is composed of an encoder and a decoder. The encoder compresses the data from the input layer into a short representation, which is a non-linear combination of the input elements. The decoder decompresses this representation into a new representation that closely matches the original data. The network is trained to minimize the reconstruction error (RE), which is the average squared distance between the original data and its reconstruction [52].

An LSTM-autoencoder [44] is an extension of an autoencoder for time-series data using an encoder-decoder LSTM architecture. An LSTM-autoencoder can capture the temporal dependencies among the input records by using LSTM networks as the layers of the autoencoder network.

Figure 13 shows the LSTM-autoencoder architecture. The input and output are fixed-size time series matrices. $X_{i,j} = [x_{i,j}^0, \dots, x_{i,j}^{d-1}]$ is the jth record with d attributes, T_i is the ith time series that contains w records, and w is the window size. The network output has the same dimensionality as the network input. The network is composed of two hidden layers that are LSTMs with d' units. The first LSTM layer functions as an encoder that investigates the dependencies from the input sequence and produces a complex hidden context (i.e., d' encoded time series features, where the value of d' depends on the underlying encoding used by the autoencoder). The second LSTM layer functions as a decoder that produces the output sequence, based on the learned complex context and the previous output state. The TimeDistributed layer is used to process the output from the LSTM hidden layer. This layer is a dense (fully-connected) wrapper layer that makes the network return a sequence with shape ($d * w$). The reconstruction error for this network is defined as follows [52]:

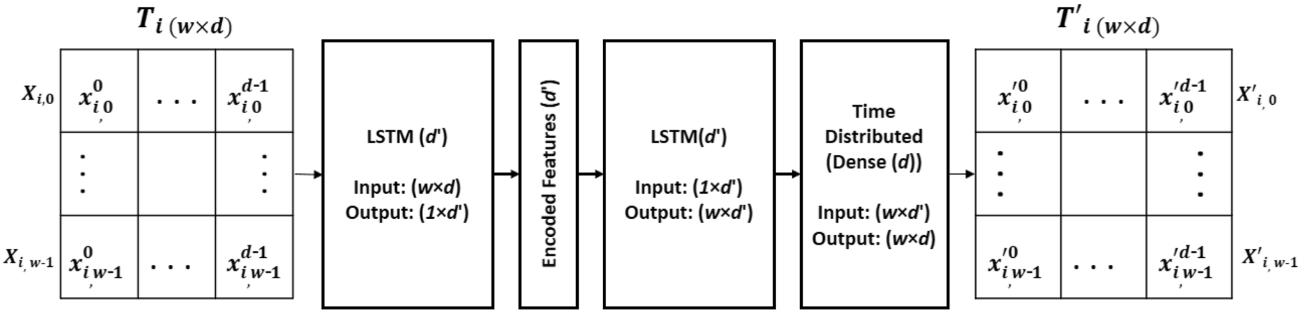


Fig. 13 An LSTM-autoencoder network

$$\mathrm{RE} = \frac{1}{m}\sum_{i=1}^{m}(T_i' - T_i)^2, \tag{6}$$

where T_i and T_i' are the ith network input and output and m is the total number of subsequences.

Acknowledgements This work was supported in part by funding from NSF under Award Numbers CNS 1650573, CNS 1822118, IIS 2027750, OAC 1931363, SecureNok, AFRL, CableLabs, Furuno Electric Company, NIST, and Google.

Declaration

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Wenham C, Smith J, Morgan R. COVID-19: the gendered impacts of the outbreak. Lancet. 2020;395(10227):846–8.
2. Jain R, Gupta M, Taneja S, Hemanth DJ. Deep learning based detection and analysis of COVID-19 on chest X-ray images. Appl Intell. 2020;51:1–11.
3. Cortegiani A, Ingoglia G, Ippolito M, Giarratano A, Einav S. A systematic review on the efficacy and safety of chloroquine for the treatment of COVID-19. J Crit Care. 2020;57:279–83.
4. Mishra T, Wang M, Metwally AA, Bogu GK, Brooks AW, Bahmani A, Alavi A, Celli A, Higgs E, Dagan-Rosenfeld O, et al. Presymptomatic detection of COVID-19 from smartwatch data. Nat Biomed Eng. 2020;4:1–13.
5. Karadayi Y, Aydin MN, Öğrencí AS. Unsupervised anomaly detection in multivariate spatio-temporal data using deep learning: early detection of COVID-19 outbreak in Italy. IEEE Access. 2020;8:164155–77. https://doi.org/10.1109/ACCESS.2020.3022366.
6. Jombart T, Ghozzi S, Schumacher D, Leclerc Q, Jit M, Flasche S, Greaves F, Ward T, Eggo RM, Nightingale E, et al. Real-time monitoring of COVID-19 dynamics using automated trend fitting and anomaly detection. medRxiv. 2020.
7. Zhu G, Li J, Meng Z, Yu Y, Li Y, Tang X, Dong Y, Sun G, Zhou R, Wang H, et al. Learning from large-scale wearable device data for predicting epidemics trend of COVID-19. Discrete Dynamics in Nature and Society. 2020;2020:6152041. https://doi.org/10.1155/2020/6152041.
8. Agbehadji IE, Awuzie BO, Ngowi AB, Millham RC. Review of big data analytics, artificial intelligence and nature-inspired computing models towards accurate detection of COVID-19 pandemic cases and contact tracing. Int J Environ Res Public Health. 2020;17(15):5330.
9. Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 2019;31(7):1235–70.
10. Kromkowski P, Li S, Zhao W, Abraham B, Osborne A, Brown DE. Evaluating statistical models for network traffic anomaly detection," in 2019 systems and information engineering design symposium (SIEDS) 2019. pp. 1–6. https://doi.org/10.1109/SIEDS.2019.8735594.
11. Wu J, Zeng W, Yan F. Hierarchical temporal memory method for time-series-based anomaly detection. Neurocomputing. 2018;273:535–46.
12. Chen Y, Wu W. Application of one-class support vector machine to quickly identify multivariate anomalies from geochemical exploration data. Geochem Explor Environ Anal. 2017;17(3):231–8.
13. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. In: ACM SIGMOD international conference on management of data. Association for Computing Machinery; 2000. p. 93–104.
14. Cheng Z, Zou C, Dong J. Outlier detection using isolation forest and local outlier factor. In: Conference on research in adaptive and convergent systems. Association for Computing Machinery; 2019. p. 161–168.
15. Thomas R, Judith JE. Voting-Based Ensemble of Unsupervised Outlier Detectors. In: Advances in Communication Systems and Networks, vol. 656, Jayakumari J, Karagiannidis GK, Ma M, Hossain SA (eds). Singapore: Springer Singapore. 2020. pp. 501–511.
16. Homayouni H, Ghosh S, Ray I. ADQuaTe: an automated data quality test approach for constraint discovery and fault detection. In: IEEE 20th international conference on information reuse and integration for data science. CA: Los Angeles; 2019. p. 61–68.
17. Homayouni H, Ghosh S, Ray I, Kahn M. An interactive data quality test approach for constraint discovery and fault detection. In: 2019 IEEE International Conference on Big Data (Big Data). 2019. pp. 200–205. https://doi.org/10.1109/BigData47090.2019.9006446.
18. Lu H, Liu Y, Fei Z, Guan C. An outlier detection algorithm based on cross-correlation analysis for time series dataset. IEEE Access. 2018;6:53593–610.
19. Homayouni H, Ghosh S, Ray I, Gondalia S, Duggan J, Kahn MG. An autocorrelation-based LSTM-Autoencoder for anomaly detection on time-series data. In: 2020 IEEE International Conference on Big Data (Big Data). 2020. pp. 5068–5077. https://doi.org/10.1109/BigData50022.2020.9378192.

20. Yahoo Server Traffic: A Benchmark Dataset for Time Series Anomaly Detection. https://yahooresearch.tumblr.com/post/114590420346/a-benchmark-dataset-for-time-series-anomaly. Accessed 1 July 2020.
21. NASA Shuttle. https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle). Accessed 15 May 2020.
22. Energy Data. https://energy.colostate.edu/. Accessed 03 May 2020.
23. Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by Johns Hopkins CSSE. https://github.com/CSSEGISandData/COVID-19. Accessed 25 Aug 2020.
24. An Ongoing Repository of Data on Coronavirus Cases and Deaths in the US. https://github.com/nytimes/covid-19-data. Accessed 25 Aug 2020.
25. The COVID Tracking Project. https://github.com/COVID19Tracking. Accessed 25 Aug 2020.
26. HDC. http://www.ucdenver.edu/about/departments/healthdatacompass/. Accessed 20 Nov 2020.
27. Shriram S, Sivasankar E. Anomaly detection on shuttle data using unsupervised learning techniques. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). 2019. pp. 221–225. https://doi.org/10.1109/ICCIKE47802.2019.9004325.
28. Hyndman RJ, Wang E, Laptev N. Large-Scale unusual time series detection. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW). 2015. pp. 1616–1619. https://doi.org/10.1109/ICDMW.2015.104.
29. Laptev N, Amizadeh S, Flint I. Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA. 2015. pp. 1939–1947. https://doi.org/10.1145/2783258.2788611.
30. Kuster C, Rezgui Y, Mourshed M. Electrical load forecasting models: a critical systematic review. Sustain Cities Soc. 2017;35:257–70.
31. Maçaira PM, Thomé AMT, Oliveira FLC, Ferrer ALC. Time series analysis with explanatory variables: a systematic literature review. Environ Model Softw. 2018;107:199–209.
32. Hasani Z, Jakimovski B, Velinov G, Kon-Popovska M. An adaptive anomaly detection algorithm for periodic data streams, “intelligent data engineering and automated learning”. Berlin: Springer International Publishing; 2018. p. 385–97.
33. Adhikari R, Agrawal RK. An introductory study on time series modeling and forecasting. Saarbrücken: LAP LAMBERT Academic Publishing; 2013.
34. Bishop CM. Bishop PoNCCM. Neural networks for pattern recognition. Oxford: Clarendon Press; 1995.
35. Park D, Hoshi Y, Kemp CC. A multimodalanomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. IEEE Robot Autom Lett. 2018;3(3):1544–51.
36. Wang B, Wang Z, Liu L, Liu D, Peng X. Data-Driven anomaly detection for UAV sensor data based on deep learning prediction model. In: 2019 Prognostics and System Health Management Conference (PHM-Paris). 2019. pp. 286–290. https://doi.org/10.1109/PHM-Paris.2019.00055.
37. Zhang W, Du T, Wang J. Deep learning over multi-field categorical data. In: Ferro N, Crestani F, Moens MF, Mothe J, Silvestri F, Di Nunzio GM, Hauff C, Silvello G (eds). Advances in Information Retrieval, vol. 9626, Cham: Springer International Publishing. 2016. pp. 45–57.
38. Shalabi LA, Shaaban Z. Normalization as a preprocessing engine for data mining and the approach of preference matrix. In: 2006 International Conference on Dependability of Computer Systems. 2006. pp. 207–214. https://doi.org/10.1109/DEPCOS-RELCOMEX.2006.38.
39. Park KI. Fundamentals of probability and stochastic processes with applications to communications. 1st ed. Berlin: Springer Publishing Company Incorporated; 2017.
40. Loganathan G, Samarabandu J, Wang X. Sequence to sequence pattern learning algorithm for real-time anomaly detection in network traffic. In: 2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE). 2018. pp. 1–4. https://doi.org/10.1109/CCECE.2018.8447597.
41. de Laat PB. Algorithmic decision-making based on machine learning from big data: can transparency restore accountability. Philos Technol. 2018;31(4):525–41.
42. Kaminski B, Jakubczyk M, Szufel P. A framework for sensitivity analysis of decision trees. Cent Eur J Oper Res. 2018;26(1):135–59.
43. Talagala PD, Hyndman RJ, Smith-Miles K, Kandanaarachchi S, Munoz MA. Anomaly detection in streaming nonstationary temporal data. J Comput Graph Stat. 2019;29:1–21.
44. Kieu T, Yang B, Jensen CS. Outlier detection for multidimensional time series using deep neural networks. In: 2018 19th IEEE International Conference on Mobile Data Management (MDM). 2018. pp. 125–134. https://doi.org/10.1109/MDM.2018.00029.
45. Guo T, Xu Z, Yao X, Chen H, Aberer K, Funaya K. Robust online time series prediction with recurrent neural networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). 2016. pp. 816–825. https://doi.org/10.1109/DSAA.2016.92.
46. Aljbali S, Roy K. Anomaly detection using bidirectional LSTM. In: Arai K, Kapoor S, Bhatia R, editors. Intelligent systems and applications. Berlin: Springer International Publishing; 2021. p. 612–9.
47. Manaswi NK. RNN and LSTM. Berkeley: Apress; 2018. p. 115–126.
48. Finlay R, Fung T, Seneta E. Autocorrelation functions. Int Stat Rev. 2011;79(2):255–71.
49. Pivot Transformation. https://cloud.google.com/dataprep/docs/html/Pivot-Transform_57344645. Accessed 18 Nov 2020.
50. Wissel BD, Van Camp PJ, Kouril M, Weis C, Glauser TA, White PS, Kohane IS, Dexheimer JW. An interactive online dashboard for tracking COVID-19 in US counties, cities, and states in real time. J Am Med Inform Assoc. 2020;27(7):1121–5.
51. Understanding LSTM Networks, Recurrent Neural Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed 29 Mar 2021.
52. Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA. 2017. pp. 665–674. https://doi.org/10.1145/3097983.3098052.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.